INTERNET PROTOCOL SECURITY FRAMEWORK UTILIZING PREDICTIVE SECURITY ASSOCIATION RE-NEGOTIATION

FIELD OF THE INVENTION

The present invention relates generally to the field of securing data using the Internet Protocol Security (IPSEC) framework as proposed by the Internet Engineering Task Force (IETF).

BACKGROUND OF THE INVENTION

To secure data over the Internet, the Internet Engineering Task Force (IETF) has recommended a set of protocols for the Internet Protocol (IP). These suites of secure protocols are referred to as Internet Protocol Security (IPSEC) protocols. IPSEC is a developing standard for security at the network or packet processing layer of network communication. Earlier security approaches had inserted security at the application layer of the communications model. IPSEC is especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks. A significant advantage of IPSEC is that security arrangements can be handled without requiring changes to individual user computers.

The IPSEC protocols rely on keys to encrypt and decrypt the data. Two parties wishing to exchange data securely using IPSEC exchange IPSEC keys between them. The secure exchange of IPSEC keys is a major factor in determining the security and the integrity of a whole system. Other factors include the strength of crypto-algorithm (DES, 3DES), procedures, etc.

For large scale deployment of IPSEC and automatic exchange of keys between parties the IETF has defined a key exchange protocol known as the IKE (Internet Key Exchange). The IKE allows two parties to exchange IPSEC keys securely and automatically over the Internet. The IPSEC keys are exchanged by IKE by negotiating Security Associations (SA's) between the two parties. Security Associations (SA's) are simplex connections that afford security services to the traffic being carried. In other words, two sides wishing to communicate using IPSecurity (as defined by the IETF)

negotiate and have Security Associations among them. The SA's specify the security parameters that should be used to communicate with the other party. For bi-directional communication, each party typically has two SA's – incoming and outgoing. For added security (to avoid key compromise) and to prevent crypto analysis of the data transferred, RFC 2401 (the IPSEC RFC), recommends that an SA be valid for only a short period of time (e.g. 20 minutes) and that new keys should be exchanged at regular intervals. Accordingly, two parties need to renegotiate another set of security associations (SA's) if they wish to continue the exchange of data after the previous SA expires. The IPSEC RFC recommends two types of units to specify the life of the SA, i.e., time and/or bytes of data transferred. Thus, if the SA life is specified as 10 Mbytes then the two parties can exchange up to 10 Mbytes of data using the current SA. To send more data, the two parties should negotiate another set of SA's for every 10 Mbytes of data.

RFC 2401 specifies the SA life in time and bytes. When a SA life is specified in time units, in order to continue to send data, an initiator system has to renegotiate another set of SA's after the SA lifetime expires. While a new SA is being renegotiated, no data can flow. To prevent data flow interruption, often a system designer anticipates the expiration of a current SA. Before the current SA expires, the initiator system starts renegotiation of new SA's such that new SA's are available as soon as the current SA's expire. This prevents data flow interruptions.

The lifespans of SA's based on time units are relatively easy to renegotiate in advance. This is because the system designer can safely assume the time it might take to negotiate a set of SA's. Based on the time to renegotiate a new SA and the time left before the old SA expires, the system designer can compute the time the system can start new SA negotiations and thus prevent data interruptions. For example, if a current SA expires at T seconds and if it takes 15 seconds to negotiate a set of SA (worst case), then the system can start renegotiation T-15 seconds before the current SA expires and thus preventing data loss/interruptions.

When SA's are specified with life units based on bytes, it is not easy for a system to predict when the SA is going to expire. This is because the data flow is not always uniform. The Internet data flow is bursty in nature. That is, there could be a burst of data flow between the two systems followed by a lull and another burst. Predictability is

extremely important in high-speed data communication systems where any interruption in the flow of data occurring due to SA re-negotiation can cause loss of lot of data. A need therefore exists to accurately predicting the expiry of SA's based in bytes.

SUMMARY OF THE INVENTION

The present invention is a methodology for predicting when current sets of encryption keys used in a high speed data network are about to expire. The invention allows network elements of a communication system to re-negotiate new sets of keys well in advance so as to prevent interruptions in communications traffic flow.

In accordance with one exemplary embodiment of the invention, a weighted traffic flow per usage for a given network element is calculated on a periodic basis. The value of the weighted traffic flow per usage is compared with a remainder value of a specific quantity of communications traffic yet to be processed by the network element. If the remainder value is less than the weighted traffic flow value, an indication is given to the appropriate network element to renegotiate a new set of keys.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention may be obtained from consideration of the following detailed description of the invention in conjunction with the drawing, with like elements referenced with like references, in which:

FIG. 1 is an exemplary embodiment of a communication network which makes use of the SA predictive coding algorithm according to the present invention;

FIG. 2 is an exemplary flow diagram illustrating computation of the SA predictive coding algorithm according to the present invention; and

FIG. 3 is a graphic and accompanying table illustrating an exemplary communications traffic flow for use in accordance with the present invention.

DETAILED DESCRIPTION

Fig. 1 shows two computer systems which couple to one another for communications purposes over the public Internet. Although the present invention is illustrated in the context of a connection over the public Internet, it would be understood that the present invention could be utilized to enhance secure communications connections over substantially any type of communications network. In the exemplary embodiment of Fig. 1, a first endpoint computer system (system A) 10 and a second endpoint computer system (system B) 20 are configured to send data securely using IPSEC over the Internet communications network 30. As discussed in the background, IPSEC is a developing standard for security at the network or packet processing layer of network communication. IPSEC is especially useful in the implementation of virtual private networks and for remote user access through dial-up connection to private networks. A significant advantage of IPSEC is that security arrangements can be handled without requiring changes to individual user computers.

In the embodiment of Fig. 1, system A 10 wishes to send communications traffic to system B 20. Accordingly, system A 10 is considered to be the initiator and system B 20 is considered to be the responder. In accordance with the subject matter of the present invention, the responder (system B) 20 has been configured to negotiate IPSEC keys with a limit, for example, of 100 Mbytes. As the communication progresses, the initiator and the responder negotiate and exchange a set of keys with a limit of 100 Mbytes of data. The keys are discarded once the limit is reached. Thus, if system A 10 wants to continue sending more data to system B 20 beyond the 100 Mbyte limit, then system A 10 has to renegotiate another set of keys with system B 20 . This allows system A to send the next 100 Mbytes. It is assumed for the purposes of this discussion that both systems A and B are systems that can renegotiate new keys without causing any interruptions in the traffic flow. Although highly desirable, such capability is not necessary for implementation of the present invention.

For security associations (SA's) limited by an amount of traffic, e.g. bytes, a predictive algorithm in accordance with the present invention is used to evaluate when a new SA should be negotiated in order to avoid an interruption in data flow. A significant advantage of the present invention is that it is accurate and simple to implement without

affecting performance of the system. As had been discussed in the background, due to the bursty nature of Internet traffic, it is not enough to compute the average flow of bytes for a given time period. The average method calculates the average number of bytes that were processed by a SA per period. For example, if for the SA during period T1 10 Mbytes of data was processed and during period T2 40 Mbytes of data was processed, then the average data processed per period is (10 + 40)Mbytes /2 periods = 25 Mbytes. This is different than an improved measurement technique which is presented in accordance with the present invention.

The improved measurement technique according to the present invention is to compute the average traffic processed per SA usage for a given time period. This is also called weighted traffic flow per usage. This is done by keeping track of how often the SA was used for a given time period and how many bytes were processed in the same period. By taking an average of the number of times the SA was accessed and the average number of bytes per usage a computer system can accurately predict when the SA will expire. This is called the weighted average of SA usage per access. Thus, with respect to the exemplary network of Fig. 1, the initiator system (e.g., system A 10), can renegotiate another set of SA's such that there are no traffic flow interruptions.

Referring to Fig. 2, an exemplary flow diagram 200 of the present invention for the calculation of the weighted average of SA usage per access is shown. As would be understood by a person skilled in the art, in an exemplary form of the invention, the negotiations would be performed by the endpoint systems, each of which includes a digital processor. As would be understood, the steps of the present invention will be embodied in software stored in memory of the endpoint systems, which is accessible by the digital processor. The invention could also be implemented in hardware, as would be understood.

In accordance with the flow diagram of Fig. 2, certain calculations are performed in accordance with the present invention methodology during every period. The calculation period is selectable according to parameters that would be known by a system's manager of a user system, for example, 15 seconds. A main criterion for selection of the time period is that the time period be smaller than the smallest known time block for transmitting the specified amount of data. This point is illustrated latter in

the application by the exemplary calculation. In general, the time period will be chosen so that at least multiple re-negotiation calculations would be accomplished during the span of the smallest known time block. An exemplary time period for a system having a 100 Mbyte SA usage limit for the exemplary system of Fig. 1 may be 15 seconds.

With respect to Fig. 2, after a suitable period has been determined, the calculation begins at the Start box 210. As a first calculation during each time period, at box 220, an average use of a given Security Association is determined. The calculation for average use of SA per period is equal to the total number of times the SA was used divided by the number of periods. The number of periods is counted from the time the SA was first negotiated. This number is updated at least at every increment in period. For example when utilizing 15 second periods, at time T0, the number of periods equals 0. After 15 seconds, the number of periods is 1 and at the end of 30 seconds is 2 and so on.

A decision box 230 is next entered to determine whether the SA has been used during the current period. If the SA was used during the period, the "Yes" path is followed to the next processing box 240. If the SA was not utilized during the current period, the "No" path is followed and the average bytes per use equal zero (box 280). The output of box 280 then loops to the input of box 250. In an alternative embodiment, the program could also loop back toward box 220 to begin another calculation of average use per period.

If the "Yes" path is followed from the decision box, the processing box 240 is entered. A calculation to determine the average number of bytes per use is performed. This value equals the number of bytes processed by SA divided by the number of times the SA was used.

Following the "Yes" path, a computation at processing box 250 is next completed to determine how much "time" remains before another SA must be negotiated. This value, referred to as "Remain" is equal to the SA life in bytes minus the number of bytes processed by the SA. The final calculation of the methodology of Fig. 2 is to determine whether the value of "Remain" is less than the average use of SA per period multiplied by the average bytes per use (value "X"). This comparison takes place at decision box 260. If the value of "Remain" is less than the average use of SA per period multiplied by the average bytes per use (value "X"), then a new SA is to be negotiated with the

responder system (box 270). On the other hand, if the value of X is greater than the value of "Remain", the SA predictor feature remains idle or sleeps until the beginning of another calculation in the same period. The calculation will also renew at the beginning of each new period.

The pseudo-code for the SA predictive renegotiation scheme is as follows:

```
In each period, compute:
  avg_use_of_SA_per_period = number of times SA was used / number of periods.
  IF SA was used then
    avg_bytes_per_use = # of bytes processed by SA / # of times SA was used.
  else
    avg_bytes_per_use = 0;

Now compute how much time before we negotiate another SA.

  remain = SA life in bytes - # of bytes processed by the SA
  IF  remain < (avg_use_of_SA_per_period * avg_bytes_per_use) then
      negotiate another SA
  ELSE
      Sleep till next time period.
```

In order to further illustrate the present invention, a sample calculation utilizing the methodology of the present invention will be explained in connection with a sample communications flow. Referring to Fig. 3, a graphic illustrating an exemplary burst traffic flow is shown for communications traffic occurring between two endpoints over three different time periods. Within the first period (end of T1), 10 Mbytes are processed. The first period (T1) is followed by a burst of 50 Mbytes during T2. T2 is followed by a lull of 10 Mbytes during T3.

Fig. 3 also illustrates the number of times that the SA is used. Note that the number of times the SA is used is the same as the number of packets processed (encrypted or decrypted) by the SA. Dividing the number of bytes processed by the SA

by the packet size derives this number. In practice, the number is updated for each packet that is processed. With regard to the instant calculation, assumptions are made for a packet size of 1000 bytes, and a SA limit of 100 Mbytes (10^6 bytes).

Taking the above information into account, it can be seen that for the sample communications flow of Fig. 3, the sample calculations utilizing the methodology of the present invention are as follows:

End of T1 Calculation:

Total Period Tp = 1, Total Bytes Tb = $10 * 10^6$, Total SA Usage Tu = $10 * 10^3$

1. Avg_use_of_SA_per_period Au = Tu / Tp = $10 * 10^3$
2. Avg_Bytes_per_use Ab = Tb / Tu = 10^3
3. Remainder, R = $100 - 10 = 90 * 10^6$
4. Since R > (1) * (2) → No SA is negotiated

End of T2 Calculations:

Total Period Tp = 2, Total Bytes Tb = $50 * 10^6$, Total SA Usage Tu = $50 * 10^3$

1. Avg_use_of_SA_per_period Au = Tu / Tp = $25 * 10^3$
2. Avg_Bytes_per_use Ab = Tb / Tu = 10^3
3. Remainder, R = $100 - 50 = 50 * 10^6$
4. Since R > (1) * (2) → No SA is negotiated

End of T3 Calculations:

Total Period Tp = 3, Total Bytes Tb = $60 * 10^6$, Total SA Usage Tu = $60 * 10^3$

1. Avg_use_of_SA_per_period Au = Tu / Tp = $20 * 10^3$
2. Avg_Bytes_per_use Ab = Tb / Tu = $3 * 10^3$
3. Remainder, R = $100 - 60 = 40 * 10^6$
4. Since R < (1) * (2) → A new SA is negotiated

Based on the above, it can be seen that a new SA is negotiated at the end of period T3. It should be noted that for the same traffic pattern, but instead using the "average bytes" method, no SA would have been negotiated at the end of T3. If in T4 period a

burst of traffic of 50Mbytes was received then the SA would expire (limit of 100Mb) and thus a new SA would have to be negotiated which would result in loss of data while a new SA is negotiated. Accordingly, a significant advantage of the present invention of prior art methodologies is illustrated.

The present invention predictive SA renegotiation algorithm is accurate in predicting the SA expire time on different types of traffic, e.g., continuous steady stream of data (constant bandwidth) and/or bursty data patterns. A unique feature of the SA predictive algorithm is its accuracy and simplicity without affecting the performance of the system. The present invention predictive algorithm is also independent of the crypto-algorithm used for encrypting the traffic.

The SA predictive algorithm can be used in all systems supporting secure traffic using IPSEC standards. The algorithm is independent of the crypto-algorithm used in encrypting the traffic itself. The algorithm is also generic such that it can be used in traffic prediction especially in burst traffic common to the Internet.

The present invention methodology has other applications of use, besides IPSEC applications over the public Internet. Examples of other possible applications include Traffic Monitoring and Network Management Applications. Traffic management applications can use the predictive algorithm to predict and identify randomly occurring patterns. For example, the number of telephone calls or highway traffic pattern. Network Management Applications can use the predictive algorithm to monitor data and predict usage of network components. For example, if a modem banks are deployed to accept calls which are arriving randomly, then using the present invention, the application can predict when the modem banks will be saturated and can automatically add additional capacity.

The foregoing description merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise various arrangements, which, although not explicitly described or shown herein, embody the principles of the invention, and are included within its spirit and scope. Furthermore, all examples and conditional language recited are principally intended expressly to be only for instructive purposes to aid the reader in understanding the principles of the invention

and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the invention, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

In the claims hereof any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a) a combination of circuit elements which performs that function or b) software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The invention as defined by such claims resides in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the claims call for. Applicant thus regards any means which can provide those functionalities as equivalent as those shown herein. Many other modifications and applications of the principles of the invention will be apparent to those skilled in the art and are contemplated by the teachings herein. Accordingly, the scope of the invention is limited only by the claims.